EXHIBIT C

Intel® Serial Flash Memory (S33)

16-, 32-, and 64-Mbit

Datasheet

Product Features

- Architecture
  - — SPI-compatible serial interface
  - — Eight 8-Kbyte parameter blocks; configurable as one 64-Kbyte main memory sector
  - — 64-Kbyte main memory sectors
    - — 16 Mbit (31 sectors)
    - — 32 Mbit (63 sectors)
    - — 64 Mbit (127 sectors)
- Voltage and Power
  - — Vcc = 2.7 V to 3.6 V
  - — Standby current: 15 µA (Typ)
  - — Read current: 3 mA (Typ) at 33.3 MHz; 6 mA (Typ) at 68 MHz
- Performance
  - — 68-MHz fast read; 33.3 MHz standard read
  - — 256-byte program buffer
- Software
  - — Driver and file manager
- Security
  - — One-Time Programmable Space
    - —64 unique factory device identifier bits
    - —64 user-programmable OTP bits
    - —Additional 3920 user-programmable OTP bits
- Quality and Reliability
  - — Operating temperature: -40 °C to +85 °C
  - — 100K minimum erase cycles per sector (block)
  - — 0.13 µm ETOX™ VIII process
- Density and Packaging
  - — 16, -32, -64-Mbit densities in SOIC-16 package
  - — 16-Mbit density in SOIC-8 package
  - — Industry standard packaging and pinout

Order Number: 314822-001US
August 2006

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT. Intel products are not intended for use in medical, life saving, life sustaining, critical control or safety systems, or in nuclear facility applications.

Intel may make changes to specifications and product descriptions at any time, without notice.

Intel Corporation may have patents or pending patent applications, trademarks, copyrights, or other intellectual property rights that relate to the presented subject matter. The furnishing of documents and other materials and information does not provide any license, express or implied, by estoppel or otherwise, to any such patents, trademarks, copyrights, or other intellectual property rights.

Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

Intel processor numbers are not a measure of performance. Processor numbers differentiate features within each processor family, not across different processor families. See http://www.intel.com/products/processor_number for details.

The Intel® Serial Flash Memory (S33) device may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an order number and are referenced in this document, or other Intel literature may be obtained by calling 1-800-548-4725 or by visiting Intel's website at http://www.intel.com.

BunnyPeople, Celeron, Celeron Inside, Centrino, Centrino logo, Core Inside, Dialogic, FlashFile, i960, InstantIP, Intel, Intel logo, Intel386, Intel486, Intel740, IntelDX2, IntelDX4, IntelSX2, Intel Core, Intel Inside, Intel Inside logo, Intel. Leap ahead., Intel. Leap ahead. logo, Intel NetBurst, Intel NetMerge, Intel NetStructure, Intel SingleDriver, Intel SpeedStep, Intel StrataFlash, Intel Viiv, Intel vPro, Intel XScale, IPLink, Itanium, Itanium Inside, MCS, MMX, Oplus, OverDrive, PDCharm, Pentium, Pentium Inside, skoool, Sound Mark, The Journey Inside, VTune, Xeon, and Xeon Inside are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

*Other names and brands may be claimed as the property of others.

Contents

Figures

Tables

Revision History

Date	Revision	Description
August 2006	001	Initial release

1.0 Introduction

This document describes the Intel® Serial Flash Memory (S33) device features, operation, and specifications.

The S33 device provides superior performance with enhanced security features, taking advantage of the high quality and reliability of the NOR-based Intel® 0.13 µm ETOX™ VIII process technology. Offered in 16-Mbit, 32-Mbit, and 64-Mbit densities, the S33 device is hardware and software compatible with existing industry offerings to ensure ease of design.

The S33 device takes advantage of more than one billion units of flash manufacturing experience and is ideal for code and data applications where simplified interface and low cost are the primary requirements. Examples include PCs and notebooks, WLAN and DSL cards and routers, printers, TVs, DVD/CD players and recorders, and other consumer electronic devices.

1.1 Nomenclature

Block:	A group of flash cells that share common erase circuitry and erase simultaneously. For Serial Flash devices, Blocks are more commonly referred to as Sectors.
Cleared:	Indicates a logic zero (0).
Page:	A 256-byte main memory segment that is aligned to a 256-byte boundary.
Parameter Block:	An 8-KB memory segment that can be erased independently.
Program:	A data write operation to the flash array.
Sector:	A single 64-KB main block, or all eight of the 8-KB parameter blocks combined.
Set:	Indicates a logic one (1).

1.2 Acronyms

LSB:	Least Significant Bit
MSB:	Most Significant Bit
OTP:	One-Time Programmable
PR:	Protection Register
PRD:	Protection Register Data
RFU:	Reserved for Future Use
SR:	Status Register
WEL:	Write Enable Latch
WSM:	Write State Machine

1.3 Conventions

0x:	Hexadecimal prefix
0b:	Binary prefix
K:	1,000
M:	1,000,000
Nibble:	4 bits
Byte:	8 bits
Word:	16 bits
Kword:	1,024 words
Kb:	1,024 bits
KB:	1,024 bytes
Mb:	1,048,576 bits
MB:	1,048,576 bytes
Brackets:	Square brackets ([]) will be used to designate group membership or to define a group of signals with similar function (i.e. A[21:1], SR[4,1] and D[15:0]).
00FFh:	Denotes 16-bit hexadecimal numbers
00FF 00FFh:	Denotes 32-bit hexadecimal numbers

2.0 Functional Overview

This section provides an overview of the features and capabilities of the Intel® Serial Flash Memory (S33) device.

The S33 device is available in 16-, 32-, and 64-Mbit densities with a common SPI interface. The SPI interface consists of 8 pins. Six of these pins are signals; the other two are Vcc and ground.

The S33 device contains eight 8-Kbyte parameter blocks and up to 127 64-Kbyte main memory sectors. The eight 8-Kbyte parameter blocks can be treated as one 64-Kbyte main memory sector.

The S33 device includes new security features. Two 8-byte, thirty 16-byte, and one 10-byte individually-lockable OTP Protection Registers can support multiple uses, including unique flash device identification.

A sector erase operation erases one of the device's 64-Kbyte main memory sectors or all eight 8-Kbyte parameter blocks in parallel, independent of other memory sectors. A parameter block erase operation erases the specified parameter block; the operation is ignored when the address is outside the parameter block space. Each block or memory sector can be independently erased 100,000 times. A bulk erase operation erases the entire chip.

Each device incorporates a write buffer of 256 bytes (128 words) to allow optimum programming performance. Page program operation uses the write buffer to program up to 256-bytes within a 256-byte-aligned main memory region. A given page can be programmed multiple times between erase cycles.

2.1 Memory Maps

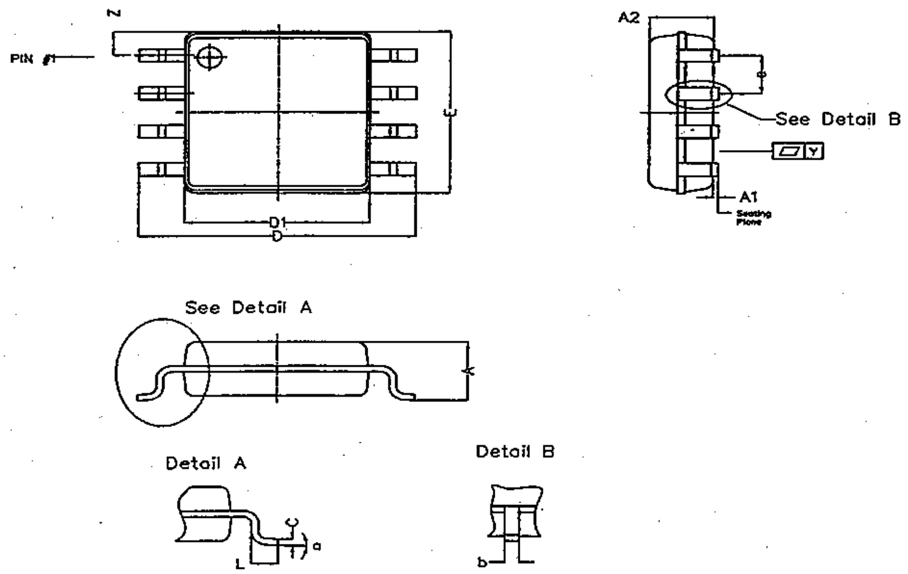
Table 1. Bottom Boot Memory Map

Size (KB)	Sector	Memory Addressing (Hex)	Memory Addressing (Hex)	Memory Addressing (Hex)
		64-Mbit	**32-Mbit**	**16-Mbit**
64	127	7F0000 - 7FFFFF		
64	126	7E0000 - 7EFFFF		
...	...	...		
64	64	400000 - 40FFFF		
64	63	3F0000 - 3FFFFF	3F0000 - 3FFFFF	
64	62	3E0000 - 3EFFFF	3E0000 - 3EFFFF	
...	...	...	...	
64	32	200000 - 20FFFF	200000 - 20FFFF	
64	31	1F0000- 1FFFFF	1F0000- 1FFFFF	1F0000- 1FFFFF
64	30	1E0000 - 1EFFFF	1E0000 - 1EFFFF	1E0000 - 1EFFFF
...	...	...	...	...
64	16	100000 - 10FFFF	100000 - 10FFFF	100000 - 10FFFF
64	15	F0000 - FFFFF	F0000 - FFFFF	F0000 - FFFFF
64	14	E0000 - EFFFF	E0000 - EFFFF	E0000 - EFFFF
...	...	...	...	...
64	4	40000 - 4FFFF	40000 - 4FFFF	40000 - 4FFFF
64	3	30000 - 3FFFF	30000 - 3FFFF	30000 - 3FFFF
64	2	20000 - 2FFFF	20000 - 2FFFF	20000 - 2FFFF
64	1	10000 - 1FFFF	10000 - 1FFFF	10000 - 1FFFF
8	0-H	E000 - FFFF	E000 - FFFF	E000 - FFFF
8	0-G	C000 - DFFF	C000 - DFFF	C000 - DFFF
8	0-F	A000 - BFFF	A000 - BFFF	A000 - BFFF
8	0-E	8000 - 9FFF	8000 - 9FFF	8000 - 9FFF
8	0-D	6000 - 7FFF	6000 - 7FFF	6000 - 7FFF
8	0-C	4000 - 5FFF	4000 - 5FFF	4000 - 5FFF
8	0-B	2000 - 3FFF	2000 - 3FFF	2000 - 3FFF
8	0-A	0 - 1FFF	0 - 1FFF	0 - 1FFF

3.0 Package Information

3.1 SOIC-8 Package

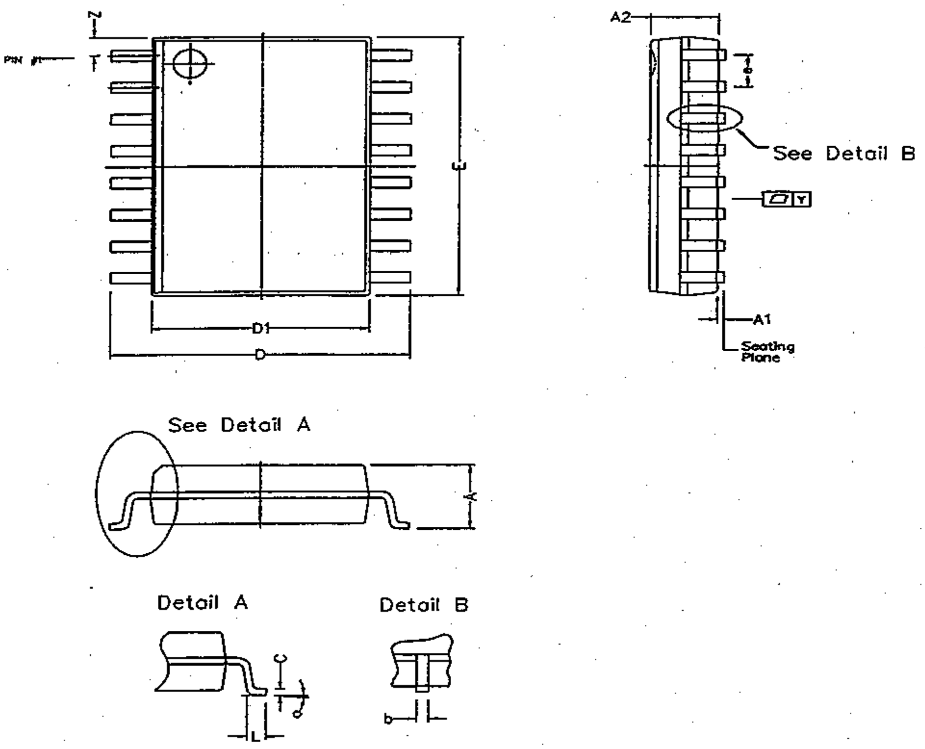
Figure 1. SOIC-8 Package Drawing and Specifications



Product Information	Symbol	Millimeters			Inches		
		Min	Nom	Max	Min	Nom	Max
Package Height	A	-	-	2.159	-	-	0.085
Standoff	A1	0.051	-	-	0.002	-	-
Package Body Thickness	A2	1.702	1.803	1.905	0.067	0.071	0.075
Lead Width	b	0.356	0.419	0.483	0.014	0.017	0.019
Lead Thickness	c	0.191	0.216	0.241	0.008	0.009	0.010
Package Body Length	D1	5.131	5.232	5.334	0.202	0.206	0.210
Package Body Width	E	5.131	5.232	5.334	0.202	0.206	0.210
Lead Pitch	e	-	1.270	-	-	0.050	-
Terminal Dimension	D	7.762	7.912	8.062	0.306	0.312	0.318
Lead Tip Length	L	0.508	0.635	0.762	0.020	0.025	0.030
Lead Count	N	-	8	-	-	8	-
Lead Tip Angle	α	0°	4°	8°	0°	4°	8°
Seating Plane Coplanarity	Y	-	-	0.100	-	-	0.004

3.2 SOIC-16 Package

Figure 2. SOIC-16 Package Drawing and Specifications



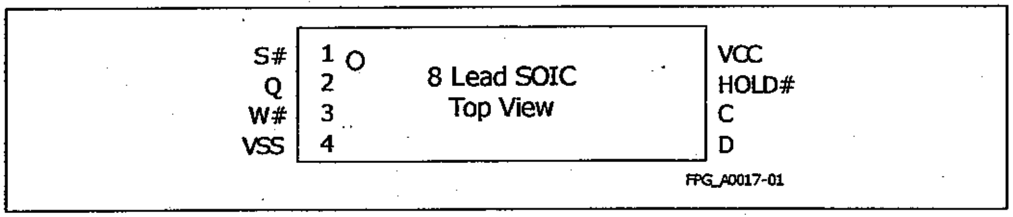
Product Information	Symbol	Millimeters			Inches		
		Min	Nom	Max	Min	Nom	Max
Package Height	A	-	-	2.642	-	-	0.104
Standoff	A1	0.102	-	-	0.004	-	-
Package Body Thickness	A2	2.286	2.362	2.438	0.090	0.093	0.096
Lead Width	b	0.315	0.412	0.508	0.012	0.016	0.020
Lead Thickness	c	0.231	0.274	0.318	0.009	0.011	0.013
Package Body Length	D1	7.442	7.519	7.595	0.293	0.296	0.299
Package Body Width	E	10.262	10.338	10.414	0.404	0.407	0.410
Lead Pitch	e	-	1.270	-	-	0.050	-
Terminal Dimension	D	10.190	10.340	10.490	0.401	0.407	0.413
Lead Tip Length	L	0.406	0.648	0.889	0.016	0.026	0.035
Lead Count	N	-	16	-	-	16	-
Lead Tip Angle	a	0°	4°	8°	0°	4°	8°
Seating Plane Coplanarity	Y	-	-	0.100	-	-	0.004

4.0 Pinouts and Signal Descriptions

The Intel® Serial Flash Memory (S33) device is available in two package types. The 16-Mbit density is supported with SOIC-8 and SOIC-16 packages as shown in Figure 3 and Figure 4. The 32- and 64-Mbit densities are supported by the SOIC-16 package as shown in Figure 4.

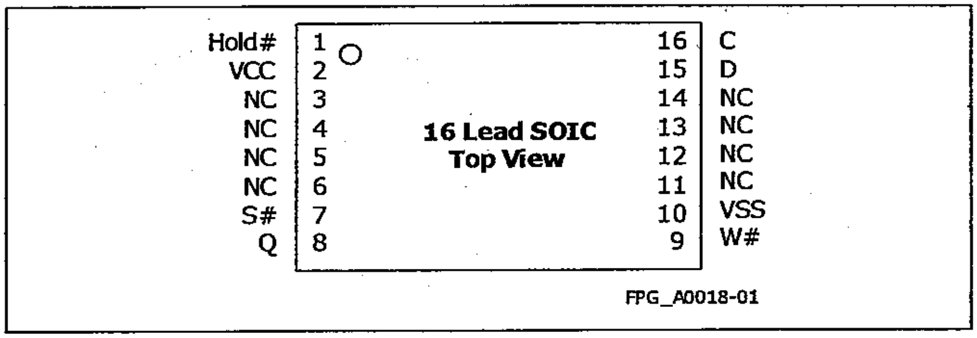
4.1 SOIC-8 Pinout (16-Mbit)

Figure 3. SOIC-8 Pinout



4.2 SOIC-16 Package Pinout

Figure 4. SOIC-16 Package Pinout



4.3 Signal Descriptions

Table 2. Signal Descriptions

Symbol	Type	Name and Function
C	Input	**SPI Clock:** Provides the timing of the SPI interface. OP codes, addresses, and data are latched in on the rising edge. SPI output data transitions after the falling edge.
D	Input	**SPI Data Input:** Shifts all data (including OP codes, Address Bytes, as well as Data Bytes) into the device. All data is clocked in on the rising edge of "C", starting with the MSB. The rising edge input applies to Modes 0 & 3 as depicted in Figure 12, "Supported SPI Bus Operation Modes" on page 25.
Q	Output	**SPI Data Output:** Shifts all data out of the device. All output data is clocked out after the falling edge of "C", starting with the MSB. The falling edge output applies to Modes 0 & 3 as depicted in Figure 12 on page 25.
S#	Input	**SPI Select:** Falling S# edge triggers command writes to the SPI interface. Rising S# edge completes (or terminates) the SPI command cycle. When S# is high, "Q" is at high-Z.
HOLD#	Input	**SPI HOLD:** Internally freezes the Synchronization Clock and sets "Q" to high-Z. To enter the Hold condition, S# must be low. Refer to Section 8.1.2, "The Hold State" on page 25 for details.
W#	Input	**Write Protect:** Enables write protection. Refer to Table 18 on page 33 for details.
VCC	Power	**Power Supply:** Source voltage. Writes to the flash array are inhibited when $V_{CC} \leq V_{LKO}$. Operations at invalid V_{CC} should not be attempted.
VSS	Power	**Ground:** Connect to system ground. Do not float VSS.

5.0 Maximum Ratings and Operating Conditions

5.1 Absolute Maximum Ratings

Warning: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only.

> ***Note:*** This document contains information available at the time of its release. The specifications are subject to change without notice. Verify with your local Intel sales office that you have the latest datasheet before finalizing a design.

Table 3. Absolute Maximum Ratings

Parameter	Min	Max	Unit	Notes
Temperature under Bias Expanded (T_A, Ambient)	–40	+85	°C	4
Storage Temperature	–65	+125	°C	—
V_{CC} Voltage	–2.0	+5.6	V	1, 2
I_{SH} Output Short Circuit Current	—	100	mA	3

Notes:

1. Voltage is referenced to V_{SS}. During infrequent non-periodic transitions, the voltage potential between V_{SS} and input/output pins may undershoot to –2.0 V for periods < 20 ns or overshoot to V_{CCQ} (max) + 2.0 V for periods < 20 ns.
2. During infrequent non-periodic transitions, the voltage potential between V_{CC} and the supplies may undershoot to –2.0 V for periods < 20 ns or V_{SUPPLY} (max) + 2.0 V for periods < 20 ns.
3. Output is shorted for no more than one second. No more than one output shorted at a time
4. Temperature specified is at ambient (T_A) and not the package (T_C).

5.2 Operating Conditions

Warning: Operation beyond the "Operating Conditions" is not recommended. Extended exposure beyond the "Operating Conditions" may affect device reliability.

Table 5. Temperature and VCC Operating Condition

Symbol	Parameter	Min	Nom	Max	Unit	Notes
T_C	Operating Temperature (case)	–40	+25	+85	°C	1
V_{CC}	V_{CC} Supply Voltage	2.7	3.0	3.6	V	

Note:

1. Temperature specified is at case (T_C) and/or ambient (T_A).

5.3 Power Supply

This section provides an overview of system-level considerations with regards to the flash device. It includes a brief description of power-up, power-down and decoupling design considerations.

5.3.1 Power-Up/Down Characteristics

The device is protected against accidental block erasure or programming during power transitions. After power-up, a t_{VSL} latency is required before S# can be brought low to issue a command.

5.3.2 Power Supply Decoupling

Flash memory devices require careful power supply decoupling. The basic power supply considerations are as follows:

1. Standby current levels
2. Active current levels
3. Transient peaks produced when the device is enabled

When the device is accessed, many internal conditions change. Circuits within the device enable charge-pumps, and internal logic states change at high speed. All of these internal activities produce transient signals. Transient current magnitudes depend on the capacitive and inductive loading at the device output. Two-line control and correct de-coupling capacitor selection suppress transient voltage peaks.

Additionally, for every eight devices used in the system, a 4.7 µF electrolytic capacitor should be placed between power and ground close to the devices. The bulk capacitor is meant to overcome voltage droop caused by PCB trace inductance.

6.0 Electrical Characteristics

6.1 DC Current Specifications

Table 6. DC Current Characteristics

Symbol	Parameter	Typ	Max	Unit	Test Conditions	Notes
I_{LI}	Input Load Current	—	±2	µA	$V_{CC} = V_{CC}$ Max; $V_{IN} = V_{CC}$ or V_{SS}	3
I_{LO}	Output Leakage Current	—	±2	µA	$V_{CC} = V_{CC}$ Max; $V_{IN} = V_{CC}$ or V_{SS}	—
I_{CCS}	V_{CC} Standby Current	15	70	µA	S# = V_{CC}; $V_{IN} = V_{CC}$ or V_{SS}	4,6
I_{DPD}	V_{CC} Deep Power-Down Current	15	70	µA	S# = V_{CC}; $V_{IN} = V_{CC}$ or V_{SS}	4
I_{CCR}	V_{CC} Read Current	2	4	mA	20 MHz; No Load	—
		3	5		33.3 MHz; No Load	7
		5	8		50 MHz; No Load	
		6	10		68 MHz; No Load	
I_{CCP}	Page Program Current	18	25	mA	S# = V_{CC}; 256-Byte Page	2,5
I_{CCE}	Parameter Block Erase Current	16	45	mA	S# = V_{CC}	2.5
	Sector Erase Current	16	45	mA	S# = V_{CC}	2,5
I_{CCB}	Bulk Erase Current	16	45	mA	S# = V_{CC}	2.5

Notes:

1. All currents are RMS unless noted. Typical values at typical V_{CC}, T_C = +25 °C.
2. Sampled, not 100% tested.
3. If $V_{IN} > V_{CC}$ the input load current increases to 10 µA max.
4. I_{CCS} and I_{DPD} is the average current measured over any 5 ms time interval 5 µs after a S# de-assertion.
5. I_{CCP}, I_{CCE} measured over typical or max times.
6. I_{CCS} will increase substantially if W# or HOLD# is toggled while in standby mode.
7. I_{CCR} will increase if D is toggled during read.

6.2 DC Voltage Specifications

Table 7. DC Voltage Characteristics

Symbol	Parameter	Min	Max	Unit	Test Conditions	Notes
V_{IL}	Input Low Voltage	0	0.3 * V_{CC}	V	—	1
V_{IH}	Input High Voltage	0.7 * V_{CC}	V_{CC}	V	—	1
V_{OL}	Output Low Voltage	—	0.1	V	$V_{CC} = V_{CC}$Min I_{OL} = 100 µA	
V_{OH}	Output High Voltage	V_{CC} – 0.1	—	V	$V_{CC} = V_{CC}$Min I_{OH} = –100 µA	—
V_{LKO}	V_{CC} Lockout Voltage	2.0	—	V	—	—

Note:

1. V_{IL} can undershoot to –1.0 V for periods <2 ns and V_{IH} may overshoot to a maximum of V_{CC} +1.0 V for periods < 2 ns.

6.3 Capacitance

Table 8. Device Capacitance

Symbol	Parameter[1]	Type	Max	Unit	Condition[2]
C_{IN}	Input Capacitance	6	8	pF	V_{IN} = 0.0 V V_{CC} = 0 V "or" V_{CC} =V_{CC}max
C_{OUT}	Output Capacitance	8	12	pF	V_{OUT} = 0.0 V V_{CC} = 0 V "or" V_{CC} =V_{CC}max

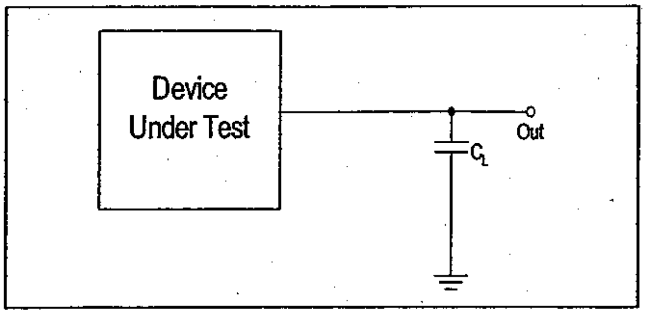
Notes:

1. Sampled, not 100% tested.
2. T_A = +25 °C, f = 1 MHz.

7.0 AC Characteristics

7.1 AC Test Conditions

Figure 5. Transient Equivalent Testing Load Circuit



Note: C_L Includes Jig Capacitance

Table 9. AC Measurement Conditions

Parameter	Min	Max	Unit
Load Capacitance (C_L)	—	30[1]	pF
Input Rise and Fall Times	0.3	3	ns
Input V^{IL} Timing Reference Voltages	—	0.3*Vcc	V
Input V^{IH} Timing Reference Voltages	0.7*Vcc	—	V
Input Drive Voltages	0 for V_{IL}, Vcc for V_{IH}		V
Input Timing Reference Voltages	0.3 V_{CC} to 0.7 V_{CC}		V
Output Timing Reference Voltages	$V_{CC}/2$		V

Note:
1. Output Hi-Z is defined as the point where data output is no longer driven.

7.2 AC Timing Characteristics

7.2.1 Serial Input Characteristics

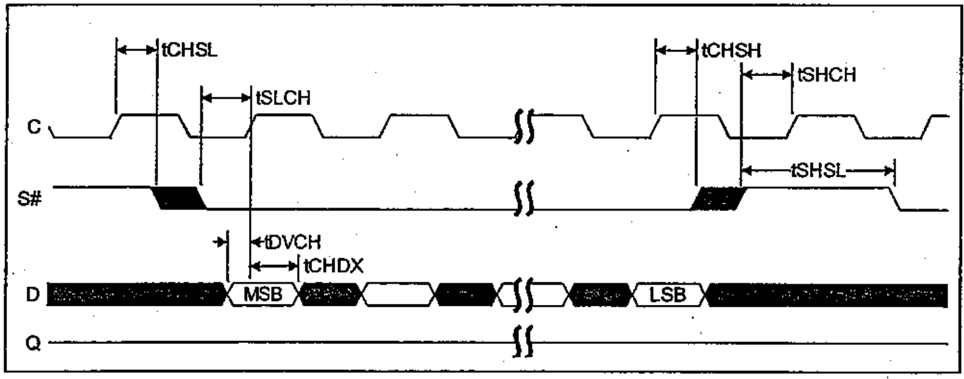
Table 10. Serial Input Characteristics

Sym	Parameter	Min	Max	Unit	Notes
f_C	Clock Frequency for all instructions except READ	D.C.	68.0	MHz	—
f_R	Clock Frequency for READ	D.C.	33.3	MHz	—
t_{CH}	Clock High Time	7	—	ns	1
t_{CL}	Clock Low Time	7	—	ns	1
t_{CLCH}	Clock Rise Time (peak to peak)	0.1	—	V/ns	2, 3,4
t_{CHCL}	Clock Fall Time (peak to peak)	0.1	—	V/ns	2, 3,4
t_{CHSL}	S# Active Setup Time (relative to preceding C edge)	5	—	ns	—
t_{SLCH}	S# Active Setup Time (relative to subsequent C edge)	5	—	ns	—
t_{CHSH}	S# Inactive Hold Time (relative to C)	5	—	ns	—
t_{SHCH}	S# Inactive Setup Time (relative to C)	5	—	ns	—
t_{SHSL}	S# Deselect Time	100	—	ns	—
t_{DVCH}	Data Input Setup Time	2	—	ns	—
t_{CHDX}	Data Input Hold Time	5	—	ns	—

Notes:
1. t_{CH} + t_{CL} must be greater than or equal to $1/f_C$(max).
2. Sampled, not 100% tested.
3. Expressed as a slew-rate.

† Minimum clock rise/fall times guarantee functionality. Clock rise/fall times must fall within the range specified in Figure 9, "AC Measurement Conditions" on page 19 for compliance to timing specs.

Figure 6. Serial Input Timing



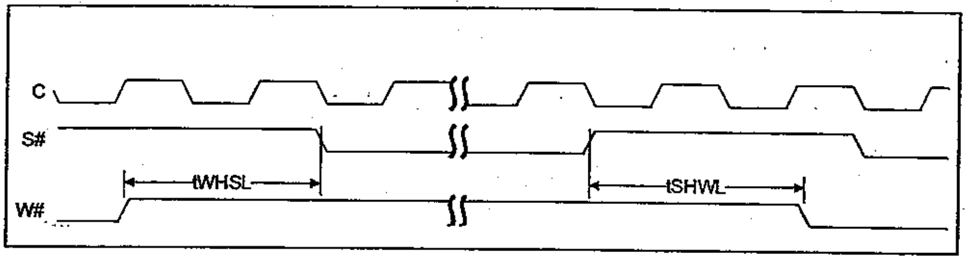
7.2.2 Write Protect Setup and Hold Timing

Table 11. Write Protect Setup and Hold Timing

Sym	Parameter	Min	Max	Unit	Notes
t_{WHSL}	W# Setup Time	20	—	ns	1
t_{SHWL}	W# Hold Time	100	—	ns	1

Note:
1. Only applicable as a constraint for a WRSR instruction when SRWD is set to 1.

Figure 7. Write Protect Setup and Hold Timing during WRSR when SRWD=1



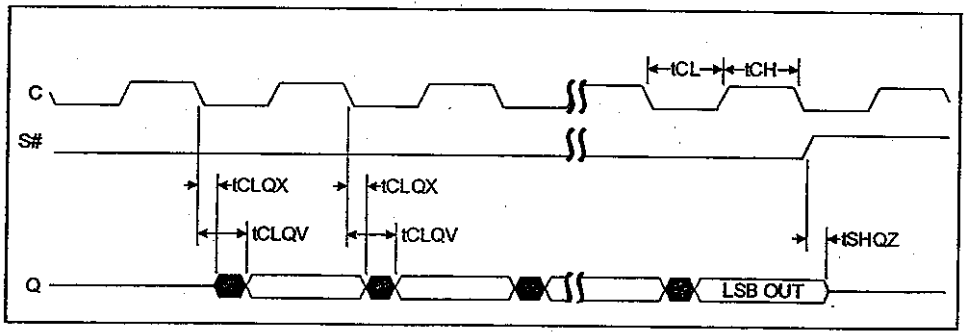
7.2.3 Output Timing

Table 12. Output Timing

Sym	Parameter	Min	Max	Unit	Notes
t_{CLQV}	Clock Low to Output Valid (30 pF, 2.7 V to 3.6 V)	—	8	ns	—
t_{CLQV}	Clock Low to Output Valid (10 pF, 3.0 V to 3.6 V)	—	6	ns	—
t_{CLQX}	Output Hold Time	0	—	ns	—
t_{SHQZ}	Output Disable Time	—	8	ns	1

Note:
1. Sampled, not 100% tested.

Figure 8. Output Timing



7.2.4 Hold Timing

Table 13. Hold Timing

Sym	Parameter	Min	Max	Unit	Notes
t_{CHHL}	HOLD# Assertion Hold Time (relative to C)	5	—	ns	—
t_{HLCH}	HOLD# Assertion Setup Time (relative to C)	5	—	ns	—
t_{HLQZ}	HOLD# Assertion to Output High-Z	—	8	ns	1
t_{CHHH}	HOLD# De-assertion Hold Time (relative to C)	5	—	ns	—
t_{HHCH}	HOLD# De-assertion Setup Time (relative to C)	5	—	ns	—
t_{HHQX}	HOLD# De-assertion to Output Low-Z	—	8	ns	1

Note:
1. Sampled, not 100% tested.

Figure 9. Hold Timing - Standard Usage

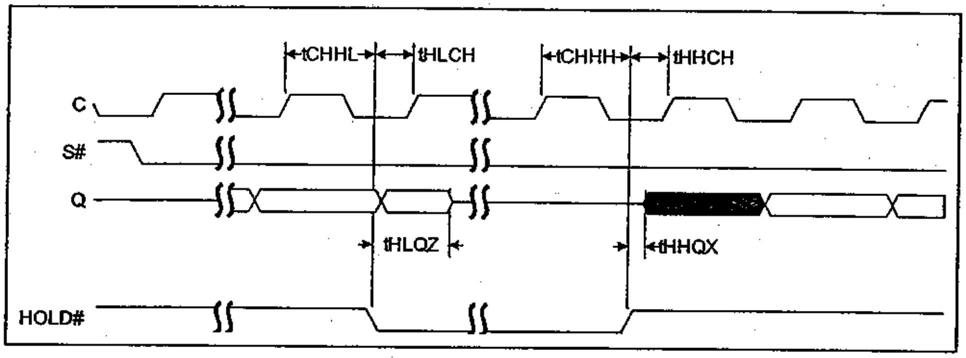
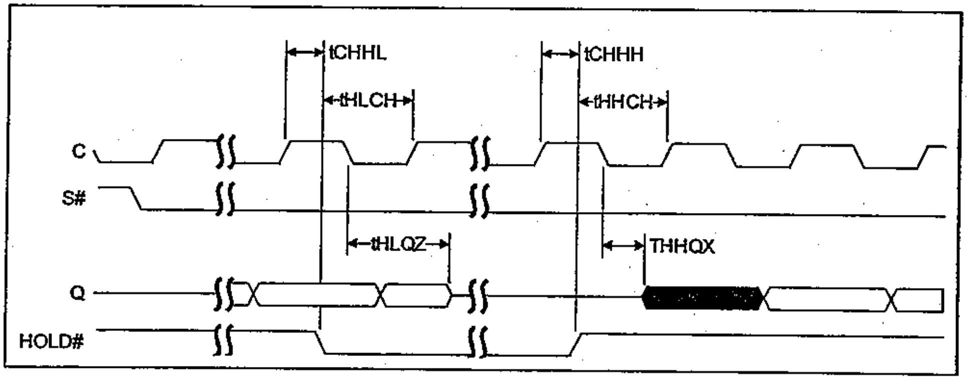


Figure 10. Hold Timing - Non-standard Usage

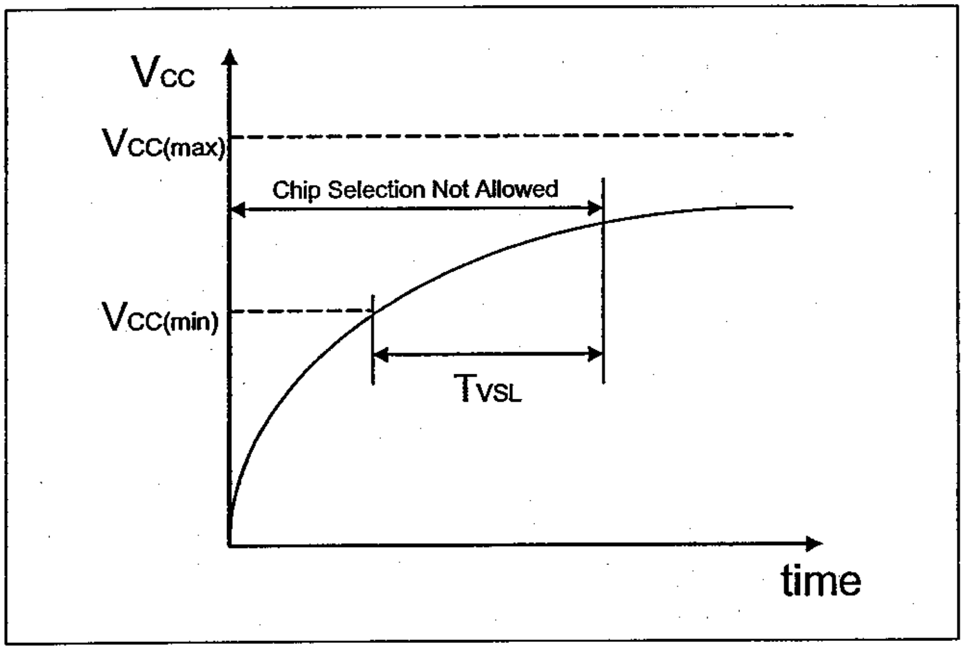


7.2.5 Other Timings

Table 14. Other Timings

Sym	Parameter	Min	Typ	Max	Unit	Notes
t_{PP}	Page Program Time	—	1.4	10	ms	1
t_{BP}	Byte OTP Program Time	—	40	175	µs	1
t_{SE}	Parameter Block Erase Time (8 KB)	—	0.3	2.5	s	1
	Sector Erase Time (64 KB)	—	0.7	4.0	s	1
t_{BE}	Bulk Erase Time (64 M)	—	89.6	512	s	1
	Bulk Erase Time (32 M)	—	44.8	256	s	1
	Bulk Erase Time (16 M)	—	22.4	128	s	1
t_{VSL}	V_{CC} power valid to S# assertion (low)	60	—	—	µs	—
t_{RDP}	Release from DPD mode into standby mode	60	—	—	µs	—

Figure 11. T_{VSL} Power-up Timing



8.0 Device Operations

This section provides an overview of the Intel® Serial Flash Memory (S33) device operations.

8.1 SPI Bus Operations

The SPI instruction cycle begins with a byte-wide OP Code that is initiated with the falling edge of S#. The 8-bit instruction is latched into "D" (data input), MSB first, on the rising edge of "C" (clock).

Some OP Codes are followed by an additional address and dummy and/or data bytes, MSB first. The number of input instruction bytes depends upon the OP Code. Refer to Table 15, "SPI Command Set" on page 26 for the instruction protocols. Address and dummy bytes are input through "D" on the rising "C" edge.

Depending upon the OP Code, the data bytes are either *input* data through "D", or they are *output* data from "Q". On cycles that input data through "D", the output signal "Q" is at high-Z.

For instructions that change the memory contents or device configuration (such as a Status Register Write command), the rising S# edge must occur on a whole-byte increment, otherwise the command will be ignored.

For read operations, the instruction sequence can be botched (ignored) only if S# is raised before the input sequence is complete. After the required number of input bits is clocked into the device, a data stream is clocked out of "Q"; each bit is shifted out after the falling edge of "C" (MSB first). When data is streaming from "Q", raising S# will terminate the data stream and bring this output to high-Z.

The rising S# edge always resets the SPI command interpreter and places the output in high-Z. It also does one of the following actions:

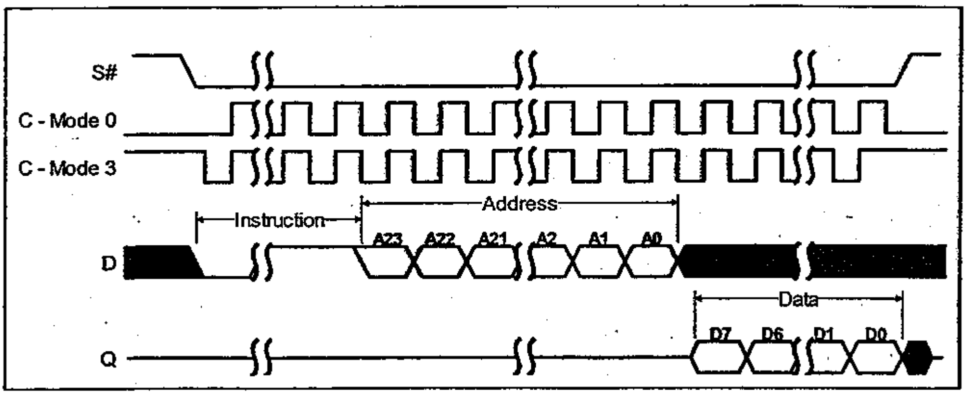
- Terminates the output data stream (read operations)
- Kicks off program/erase algorithms
- Initiates changes to the SR
- Botches an SPI command when S# is raised too early (or too late for commands that alter the array or device configuration)
- Terminates a command and puts the device in standby mode (not in the case of a program or erase operation)

When S# is high and the internal algorithms are completed, the device will go into standby mode.

8.1.1 SPI Modes

This device supports SPI bus operations Mode 0 and Mode 3, as depicted in Figure 12, "Supported SPI Bus Operation Modes" on page 25. The difference between the two modes is the default state of the clock signal (C) when the SPI bus master is in standby. For Mode 0, the "C" is normally low; for Mode 3, "C" is normally high. For both modes, input data (D) is sampled on the rising edge of "C", and output data (Q) is updated on the falling edge of "C."

Figure 12. Supported SPI Bus Operation Modes



8.1.2 The Hold State

The HOLD# input signal freezes the internal Clock (C) without resetting the device's clocking sequence. However, taking HOLD# to V_{IL} does not terminate any program or erase operation that is currently in progress.

To enter the Hold State, the device must be selected (S# at V_{IL}). The Hold State starts on the falling edge of the HOLD# signal, provided that it coincides with the Clock at V_{IL} as shown in Figure 13 on page 25. The Hold State ends on the rising edge of HOLD#, provided that it coincides with the SPI Clock at V_{IL}.

If the falling HOLD# edge does not coincide with the Clock at V_{IL}, the Hold State starts after the next falling edge of the Clock as shown in Figure 14 on page 26. Similarly, if the rising edge does not coincide with the Clock at V_{IL}, the Hold State ends after the next falling edge of the Clock.

The command sequence will not necessarily botch if S# is raised to V_{IH} while HOLD# is at V_{IL}. Raising S# will complete the command sequence, regardless of the state of HOLD#. If a proper sequence was inputted prior to dropping HOLD# to V_{IL}, the input data stream will be recognized as a valid command sequence.

During the Hold State, the Data Output (Q) is at high impedance. The Clock Input and the Data Input (D) are Don't Care.

Figure 13. Hold State — Standard Usage

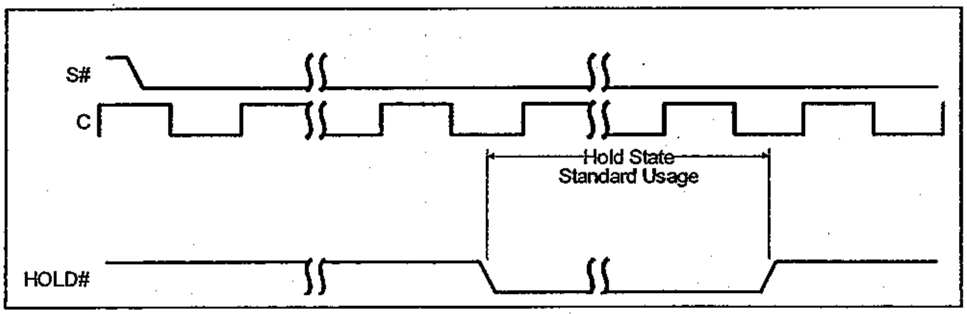
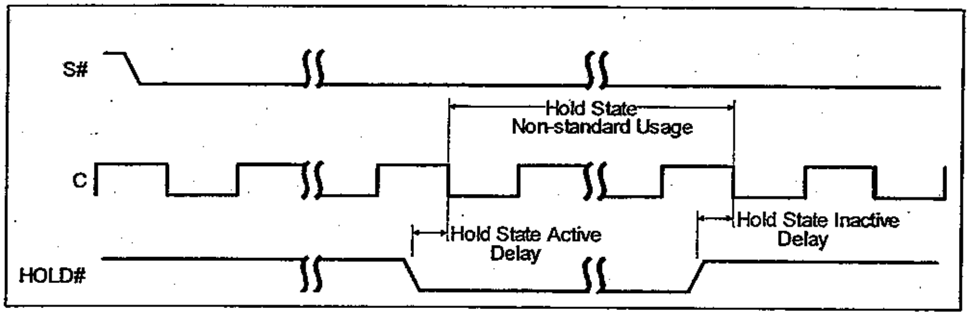


Figure 14. Hold State – Non-standard Usage



8.2 SPI Command Set

The SPI Command Set is found in Table 15. This table defines the commands (and the OP Codes) that are supported by the SPI interface. All other OP Codes will be ignored. All commands support frequencies up to 68 MHz except for the Read Data Bytes command (03h). The Read Data Bytes command (03h) supports up to 33.3 MHz.

Table 15. SPI Command Set (Sheet 1 of 2)

Instruction	Op Code	Addr Bytes	Dummy Bytes	Data Bytes	Name and Function
Write SPI SR	01h	–	–	1	Overwrites SR[7, 4:2] and (BP[2:0] SRWD). When W# = V_{IL} & SRWD = 1, the values of BP[2:0] & SRWD cannot be changed.
Page Program	02h	3	–	1 to 256	Programs up to 256 bytes within a 256-byte-aligned main memory region.
Read Data Bytes	03h	3	–	1 to infinite	Supports up to 33.3 MHz Clock.
Write Disable	04h	–	–	-	Clears the WEL bit (SR1)
Read SR	05h	–	–	1 to infinite	Continuously Polls the SR
Write Enable	06h	–	–	-	Sets the WEL bit (SR1).
Fast Read Data Bytes	0Bh	3	1	1 to infinite	Supports up to a 68 MHz Clock; protocol uses a dummy byte.
Clear SR Fail Flags	30h	–	–	–	Resets the Erase Fail Flag and the Program Fail Flag (SR5 and SR6, respectively)
Parameter Block Erase	40h	3	–	–	Erases the specified parameter block; it is ignored when the address is outside parameter block space.
OTP Program	42h	3	–	1	Programs one byte of data in the OTP memory space.
Read OTP Data Bytes	4Bh	3	1	1 to infinite	Reads data in the OTP memory space. For details refer to Section 9.1.2, "Reading OTP Data" on page 37.
Read ID	9Fh	–	–	1 to 3	Device ID: 1st Byte = MFG ID 2nd Byte = Upper Byte, 3rd Byte = Lower Byte
Release from DPD only	ABh	–	–	–	Brings the device out of DPD mode into standby mode after the t_{RDP} latency.

Table 15. SPI Command Set (Sheet 2 of 2)

Instruction	Op Code	Addr Bytes	Dummy Bytes	Data Bytes	Name and Function
Deep Power-Down	B9h	—	—	—	Puts device in DPD mode, whereby all commands are ignored except the Release from DPD command (ABh).
Bulk Erase	C7h	—	—	—	Serially erases all main memory Sectors including the eight parameter blocks.
Sector Erase	D8h	3	—	—	Erases a 64 KB Memory Sector; when addressing a parameter block, it will erase all eight 8 KB parameter blocks.

8.2.1 Write SPI SR Command (01h)

The Write SR command allows the user to write to the writable Status Register bits (that is, SR[7, 4:2]. As with any command that writes to the device, the Write Enable command must be executed prior to the Write SR command to set the WEL bit. If the WEL bit is not set, the Write SR command will be ignored.

If the device is in Hardware Protect mode, the Write SR command will be ignored and the WEL bit will be unchanged.

Assuming the WEL bit is set and the device is not in Hardware Protect mode, the rising edge of S# updates the SR within the t_{SHSL} specification time. If the Write SR command is botched (rising S# edge does not occur after exactly sixteen clock cycles), the writable SR bits and the WEL bit will remain unchanged.

8.2.2 Page Program Command (02h)

The Page Program command programs 1 bit to 256 bytes of data within a 256-Byte-Aligned memory segment. This command is used for programming the main array; it is not used for OTP programming.

The command sequence consists of an 8-bit OP Code, followed by a 24-bit address, and then by the data bytes to be programmed. The data to be programmed must be in whole-byte granularity. Otherwise, the command sequence will be ignored. To program in bit granularity, the remainder of the bits within the data byte must be set to "1".

The input data stream is loaded into a 256-Byte program buffer. The starting address of the program buffer is A[7:0] of the user-supplied address, and all subsequent bytes from the input data stream are loaded sequentially into the program buffer. If the program buffer reaches its maximum address, it wraps over, and subsequent data bytes are sequentially loaded starting at the beginning of the program buffer.

If more than 256 bytes of data are provided in the command input stream, the program buffer will be over-written, replacing the data that was previously loaded. The command sequence ends when S# goes high. When the command sequence ends, the data in the program buffer is programmed in the 256-Byte-Aligned memory segment defined by A[23:8] of the user-supplied address.

A byte count is not required with this command. The end of the data stream is identified with the rising edge of S#.

As with any command that writes to the device or changes the memory contents, the Write Enable command must be executed prior to the Page Program command in order to set the WEL bit. If the WEL bit is not set, the Page Program command will be ignored.

If the WEL bit is set and the address is protected, the program operation will not occur. Instead, the P_FAIL flag of the Status Register will set and the WEL bit will clear.

The Page Program command can be botched (cancelled) by failing to raise the S# edge on a whole-byte increment. The following occur if the Page Program command is botched:

- The WEL bit will not be cleared.
- SR Fail Flags (described in Table 16) will not be set.
- None of the bytes written into the program buffer will be programmed into the flash array.

Assuming the WEL bit is set, the address is unprotected, and the command is not botched, the rising edge of S# initiates the program operation. This program operation cannot be terminated without powering off the device, and doing so will result in unexpected data.

8.2.3 Read Data Bytes Command (03h)

The Read Data Bytes command requires a 3-byte address. After the last address byte is clocked in (on the rising clock edge), the first data bit is clocked out on the subsequent falling clock edge. Data clocks out continuously and sequentially as long as S# remains low. When the address reaches its maximum, it wraps back to zero. The Read Data Byte command supports up to 33.3 MHz.

8.2.4 Write Disable Command (04h)

The Write Disable command clears the WEL bit, which corresponds to bit SR1. Clearing the WEL bit disables the following commands:

- Write SR
- Page Program
- OTP Program
- Bulk Erase
- Parameter Block Erase
- Sector Erase

These commands can be re-enabled by executing the Write Enable command as described in Section 8.2.6. The WEL bit is cleared at power-up.

8.2.5 Read SPI SR Command (05h)

The Read SR command continuously polls the SPI Status Register. As long as S# remains low, a refreshed version of the status register is continuously clocked out. The Read SR command does not require address bytes, data bytes or dummy bytes.

8.2.6 Write Enable Command (06h)

The Write Enable command sets the WEL bit, which corresponds to bit SR1. Setting the WEL bit enables the following commands:

- Write SR
- Page Program
- OTP Program
- Bulk Erase

- Parameter Block Erase
- Sector Erase

These commands can be disabled by executing the Write Disable command as described in Section 8.2.4. The WEL bit is cleared at power-up.

8.2.7 Fast Read Data Bytes Command (0Bh)

The Fast Read Data Bytes command requires a 3-byte address. After the last address byte is clocked in (on the rising clock edge), a dummy byte latency occurs (8 clock cycles) before the first data bit is clocked out on the falling clock edge. Data clocks out continuously and sequentially as long as S# remains low. Refer to section Section 8.4.1, "Fast Read" on page 33 for a detailed description of the Fast Read Data Bytes command. When the address reaches its maximum, it wraps back to zero. The Fast Read Data Bytes command supports up to 68 MHz, but it requires a dummy byte to allow time for the first Read latency.

8.2.8 Clear SR Fail Flags Command (30h)

The Clear SR Fail Flags command resets bit SR5 (Erase Fail Flag) and bit SR6 (Program Fail Flag). It is not necessary to set the WEL bit before the Clear SR Fail Flags command is executed. The WEL bit will be unchanged after this command is executed.

8.2.9 Parameter Block Erase Command (40h)

The Parameter Block Erase command is used to erase an 8-KB Parameter block. The command sequence consists of the OP Code followed by an address within the targeted block.

As with any command that writes to the device or changes the memory contents, the Write Enable command must be executed prior to the Parameter Block Erase command in order to set the WEL bit. The Parameter Block Erase command will be ignored if the WEL bit is not set.

If the WEL bit is set and one of the following is true:

- the address is not an 8-KB Parameter Block Address
- the address is protected,

the erase operation will not occur. Instead, the E_FAIL flag of the Status Register will set and the WEL bit will clear.

The Parameter Block Erase command can be botched (cancelled) by failing to raise the S# edge after exactly thirty-two clock cycles. If the Parameter Block Erase command is botched, the WEL bit will not clear and the E_FAIL flag will not set.

Assuming the WEL bit is set, the address is an unprotected 8-KB Parameter Block address, and the command is not botched, the rising edge of S# initiates the erase operation. This erase operation cannot be terminated without powering off the device, and doing so will result in unexpected data.

8.2.10 OTP Program Command (42h)

The OTP Program command programs data in the OTP region, which is in a different address space from the main array data. Refer to Section 9.1, "OTP Memory Space" on page 37 for details on the OTP region. The protocol of the OTP Program command is the same as the Page Program command, except that the OTP Program command requires exactly one byte of data; otherwise, the command will be ignored. To program the OTP in bit granularity, the rest of the bits within the data byte can be set to "1".

If the WEL bit is set and the address is not a valid OTP address, the program operation will not occur. Instead, the F_FAIL flag of the Status Register will set and the WEL bit will clear.

The OTP memory space can be programmed one or more times, provided that the OTP memory space is not locked (as described in Section 9.1.3, "Lock Protection Registers" on page 37). Subsequent OTP programming can be performed only on the unprogrammed bits (that is, "1" data).

8.2.11 Read OTP Data Bytes Command (4Bh)

The Read OTP Data Bytes command reads data from the OTP region. Refer to Section 9.1, "OTP Memory Space" on page 37 for details on the OTP region. The protocol of the Read OTP Data Bytes command is the same as the Fast Read Data Bytes command except that it will not wrap to the starting address after the OTP address is at its maximum; instead, the data will be indeterminate.

8.2.12 Read ID Command (9Fh)

The Read ID command reads three bytes of data. It does not require any address bytes, data bytes or dummy bytes. After inputting the OP code, three bytes are clocked out of the device: the MFG ID, the first byte of the Device ID, and the second byte of the Device ID. Refer to Table 19, "Device ID Codes" on page 39 for the MFG ID and Device IDs of the S33 device. If S# is held low after the third byte of data, the subsequent data is indeterminate.

8.2.13 Release from DPD Command (ABh)

In Deep Power-Down mode, the only command that the SPI interface recognizes is the Release from DPD command. After issuing this command, the t_{RDP} latency is required before S# can go low to initiate another command. The only other way to release the device from DPD mode is to power it off and on.

When issuing a Release from DPD command, it is not necessary to raise S# after exactly eight clock cycles. After inputting the 8-bit OP Code, all subsequent bits in the command sequence will be ignored.

8.2.14 Deep Power-Down Command (B9h)

The Deep Power-Down command puts the device in DPD mode, whereby all commands are ignored except the Release from DPD command (ABh).

8.2.15 Bulk Erase Command (C7h)

The Bulk Erase command serially erases the entire main array. including the parameter blocks (but excluding the OTP memory space).

As with any command that writes to the device or changes the memory contents, the Write Enable command must be executed prior to the Bulk Erase command to set the WEL bit. The Bulk Erase command will be ignored if the WEL bit is not set.

If the WEL bit is set and there is at least one memory sector that is protected, the erase operation will not occur. Instead, the E_FAIL flag of the Status Register will set and the WEL bit will clear.

The Bulk Erase command can be botched (cancelled) by failing to raise the S# edge after exactly eight clock cycles. If the Bulk Erase command is botched, the WEL bit will not clear and the E_FAIL flag will not set.

Assuming the WEL bit is set, all memory sectors are unprotected, and the command is not botched, the rising edge of S# initiates the erase operation. This erase operation cannot be terminated without powering off the device, and doing so will result in unexpected data.

8.2.16 Sector Erase Command (D8h)

The Sector Erase command is used to erase a 64-KB memory sector. The command sequence consists of an 8-bit OP code followed by a 24-bit address. If the address is within the parameter block address range (A[max:16]=0), all eight parameter blocks will be erased. (With this characteristic, the device behaves as a symmetrically blocked device.)

As with any command that writes to the device or changes the memory contents, the Write Enable command must be executed prior to the Sector Erase command in order to set the WEL bit. If the WEL bit is not set, the Sector Erase command will be ignored.

If the WEL bit is set and the address is protected, the erase operation will not occur. Instead, the E_FAIL flag of the Status Register will set and the WEL bit will clear.

The Sector Erase command can be botched (cancelled) by failing to raise the S# edge after exactly 32 clock cycles. If the Sector Erase command is botched, the WEL bit will not clear and the E_FAIL flag will not set.

Assuming the WEL bit is set, the address is an unprotected, and the command is not botched, the rising edge of S# initiates the erase operation. The erase operation cannot be terminated without powering off the device, and doing so will result in unexpected data.

8.3 Status Register Definition

The Status Register bit definition can be found in Table 16. The Status Register has program/erase fail flags, and it contains writeable bits that define the program/erase protection within the Flash array. All Status Register bits are volatile.

Table 16. Status Register Bit Definition

SR Bit	Bit Name	Bit Description	Power-up State	Write/Read Capability
7	SRWD	**Status Register Write Disable** - When this writeable bit is set and W# is low, none of the writable SR bits can be changed including these bits (that is, SR7, 4:2). For details, refer to Table 18, "Main Array Protection Modes" on page 33. This bit is volatile.	0	Read/Write[1]
6	P_FAIL	**Program Fail Flag** - When set, this bit indicates that a program failure occurred. This bit will also be set when the user attempts to program a protected main memory region or a locked OTP region. However, the Program Fail Flag will not be set under the following scenarios: • Botched Command Sequence (that is, S# edge not raised on a whole-byte increment) • Write Enable Latch bit is reset (that is, SR1 = 0) After a series of program operations, this bit indicates whether one or more of these operations failed. Once set, this bit is reset with the Clear SR Fail Flag command.	0	Read only[2]
5	E_FAIL	**Erase Fail Flag** - When set, this bit indicates that an erase failure occurred. This bit will also be set when the user attempts to erase a protected main memory region. However, the Erase Fail Flag will not be set under the following scenarios: • Botched Command Sequence (that is, S# edge not raised on a whole-byte increment). • Write Enable Latch bit is reset (that is, SR1 = 0) After a series of erase operations, this bit indicates whether one or more of the operations failed. Once set, this bit is reset with the Clear SR Fail Flag command.	0	Read only[2]
4	BP2	**Sector Protect Bits** - These bits define the lock region of the main memory. A locked region is one or more adjacent memory sectors that are protected from program or erase. For further details, refer to Table 17. When all three of these bits are "0", the entire main array is unlocked. These bits are volatile; at power-up, these bits are set to "1".	1	Read/Write[1]
3	BP1		1	Read/Write[1]
2	BP0		1	Read/Write[1]
1	WEL	**Write Enable Latch** - This bit must be set prior to the following SPI Commands: • Write SPI SR • Page Program • OTP Program • Bulk Erase • Parameter Block Erase • Sector Erase After issuing one of these commands, the WEL bit will clear when the command is completed. The WEL bit will not be cleared if the command is botched by not raising S# on a whole-byte increment.	0	read only[3]
0	WIP	**Write in Process** - When a program, erase, or write to the SR is in process (busy), the WIP reads as "1". When the WIP bit is zero, the SPI interface is in its ready state.	0	read only

Notes:

1. Refer to Table 18, "Main Array Protection Modes" on page 33 for writable conditions.
2. The Program and Erase Fail flags are not directly writable, but they can be reset with the Clear SR Fail Flag command. This is true regardless of the protection modes described in Table 18.
3. The WEL bit is not directly writable but it can be set with the Write Enable command and reset with the Write Disable command. This is true regardless of the protection modes described in Table 18.

8.3.1 Main Memory Protection

Main memory program/erase protection is defined by three Status Register bits (SR[4:2]) and the W# input signal. Table 17 defines the memory protection regions. Table 18 defines the SR and W# configurations for Software Protection Mode and Hardware Protection Mode.

Table 17. Main Array Protection Regions - Bottom Boot

Status Register Bits			Protected Main Memory Sectors		
BP2	BP1	BP0	64 M	32 M	16 M
0	0	0	None	None	None
0	0	1	Sectors 126 - 127 (upper 1/64)	Sector 63 (upper 1/64)	Sector 31 (upper 1/32)
0	1	0	Sectors 124 - 127 (upper 1/32)	Sectors 62 - 63 (upper 1/32)	Sectors 30 - 31 (upper 1/16)
0	1	1	Sectors 120 - 127 (upper 1/16)	Sectors 60 - 63 (upper 1/16)	Sectors 28 - 31 (upper 1/8)
1	0	0	Sectors 112 - 127 (upper 1/8)	Sectors 56 - 63 (upper 1/8)	Sectors 24 - 31 (upper 1/4)
1	0	1	Sectors 96 - 127 (upper 1/4)	Sectors 48 - 63 (upper 1/4)	Sectors 16 - 31 (upper 1/2)
1	1	0	Sectors 64 - 127 (upper 1/2)	Sectors 32 - 63 (upper 1/2)	All Sectors
1	1	1	All Sectors	All Sectors	All Sectors

Table 18. Main Array Protection Modes

W# Signal	SRWD Bit (SR7)	Mode	Protection of SR
V_{IL}	0	Software Protect	The SRWD bit (SR7) and the BP bits (SR[4:2]) are writable.
V_{IL}	1	Hardware Protect	The SRWD bit (SR7) and the BP bits (SR[4:2]) are ***not*** writable. These bits cannot be altered without raising W# to V_{IH} or device power-up. In this configuration, the Main Array Protection Regions cannot be changed.
V_{IH}	0	Software Protect	The SRWD bit (SR7) and the BP bits (SR[4:2]) are writable.
V_{IH}	1	Software Protect	The SRWD bit (SR7) and the BP bits (SR[4:2]) are writable.

8.4 SPI Instruction Cycle Examples

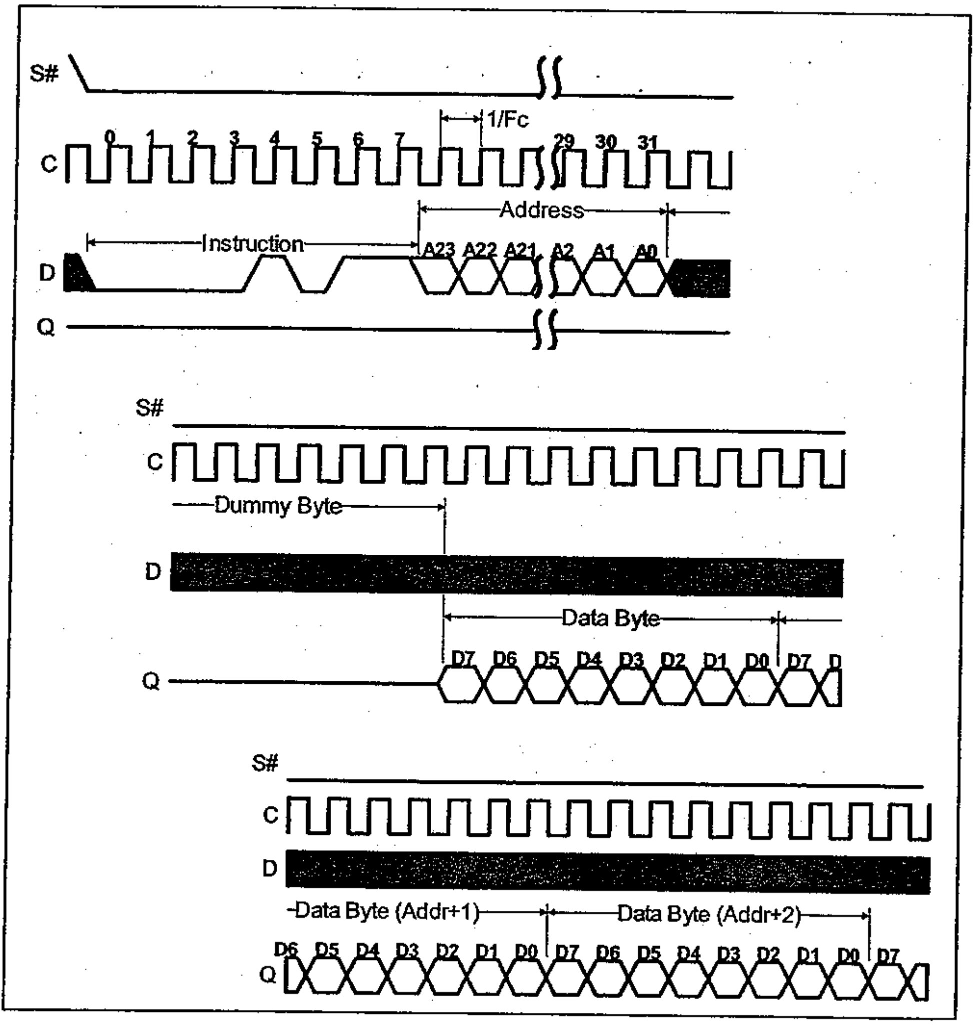
In this section, two SPI instruction cycles are explained in detail to provide a thorough understanding of SPI instruction cycles in general. The intent of these examples is to provide a foundation for all SPI instructions.

8.4.1 Fast Read

A Fast Read instruction uses 0Bh as an OP Code. After clocking in the OP Code, a 3-byte address is clocked in (starting with the MSB), followed by a dummy byte. The output serial data stream is clocked out on the falling edge of "C", one-half cycle after the last dummy bit is clocked in. The timing diagram for a Fast Read command sequence can be found in Figure 15, "Timing Diagram for SPI Fast Read Command Sequence" on page 34.

Within the SPI interface, the address is automatically incremented internally as the data is clocked out continuously and sequentially, as long as S# remains low. The output data stream can be paused by bringing HOLD# low, and it can be continued by bring it high again. When the internal address reaches the last address within the device's range, it will wrap to address 0h. When the user brings S# high, the instruction cycle is terminated, and the data output (Q) becomes tri-stated.

Figure 15. Timing Diagram for SPI Fast Read Command Sequence

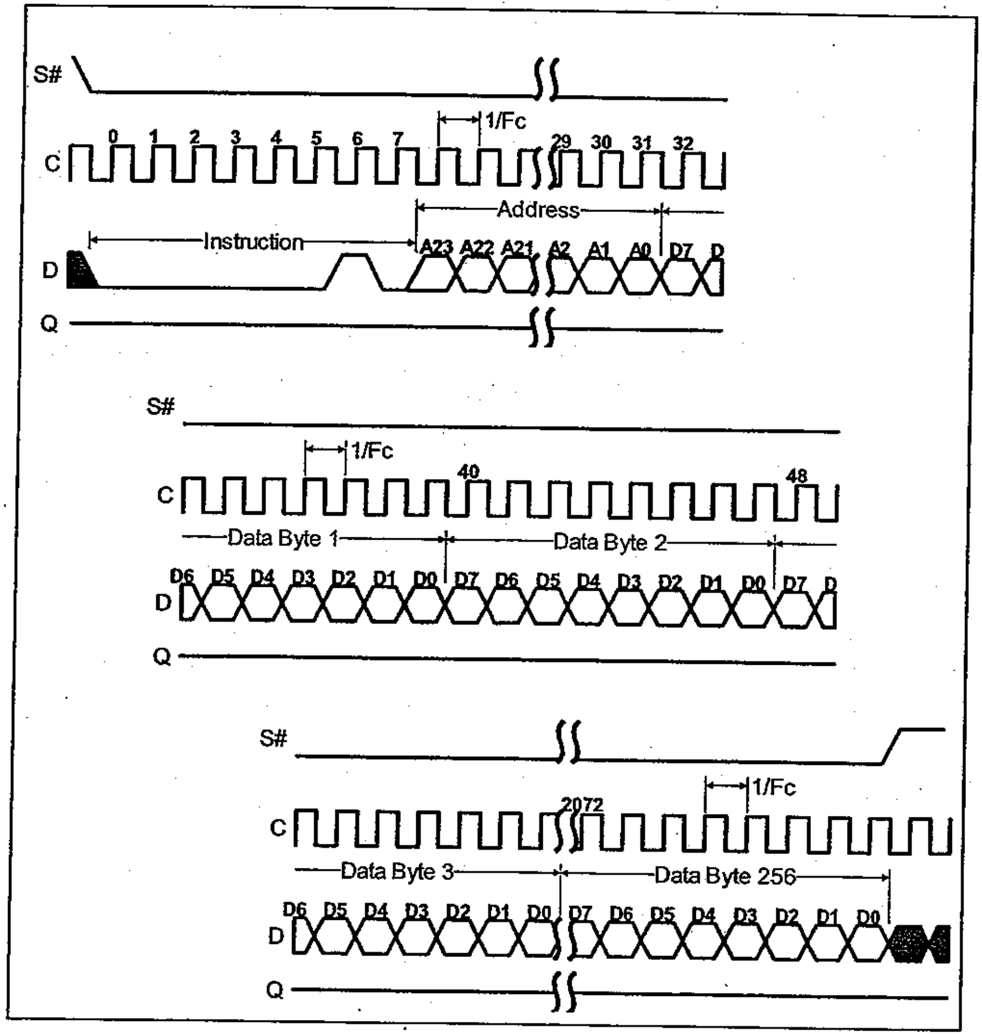


8.4.2 Page Program

A Page Program instruction consists of an OP Code (02h) followed by a 3-byte address and a variable number of data bytes, up to the size of the program buffer (page). Assuming S# goes high on a whole-byte increment, the SPI module will instruct the WSM to initiate programming, otherwise the Page Program instruction will botch (and nothing will be programmed). The timing diagram for a Page Program command sequence can be found in Figure 16, "Timing Diagram for SPI Page Program Command Sequence" on page 35.

To monitor when the program algorithm is complete, a Read SR command must be issued. The Read SR command is the only instruction that the device will recognize while a write is in process.

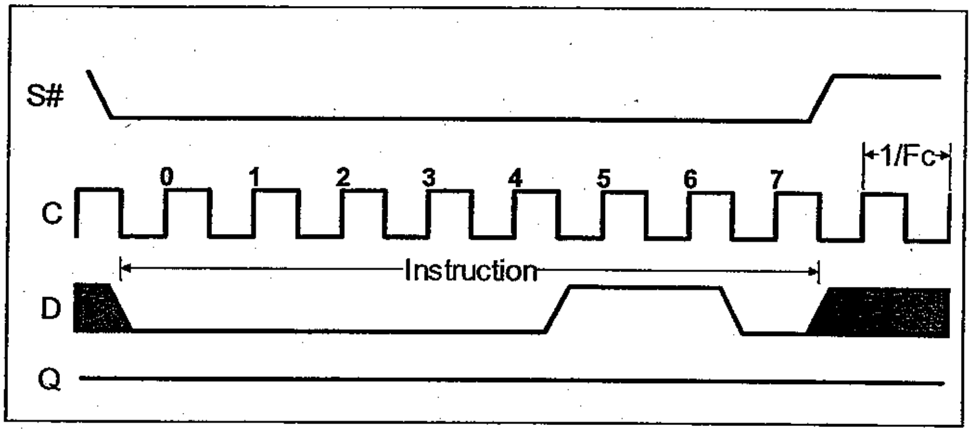
Figure 16. Timing Diagram for SPI Page Program Command Sequence



8.4.3 Write Enable

For write operations such as the Page Program operation, the single-byte Write Enable command sequence must be issued to set the WEL bit prior to issuing the Page Program operation. Without setting the WEL bit, the subsequent Page Program operation will be ignored. Just as with any other command that alters the configuration, the Write Enable command is completed on the rising edge of S#. The timing diagram for a Write Enable command sequence can be found in Figure 17, "Timing Diagram for Write Enable Command Sequence" on page 36.

Figure 17. Timing Diagram for Write Enable Command Sequence



9.0 Security Features

This section describes the security features of the S33 device that go beyond the typical SPI Flash feature set.

9.1 OTP Memory Space

The S33 device contains two 8-Byte, thirty 16-Byte, and one 10-Byte individually-lockable OTP regions (Protection Registers) within an address space that is separate from the main array. Refer to Figure 18, "OTP Memory Map" on page 38 for a pictorial representation of the OTP memory space.

The two 8-Byte Protection Registers are intended for increased system security. Protection Register values can "mate" a flash component with system CPU/ASIC to prevent device substitution. The Intel factory programs the lower 8-Byte Protection Register with a unique, unchangeable 64-bit number. The other 64 bits (upper 8-Byte) are blank so customers can program them for a similar purpose.

Once programmed, each customer segment (one of the 8-Byte segment, thirty 16-Byte segments, and one 10-Byte segment) can be locked to prevent further reprogramming.

9.1.1 Programming OTP Address Space

For the description and SPI protocol of the OTP Program command, refer to Table 15, "SPI Command Set" on page 26. The protocol of this command is the same as Page Program.

The OTP Program command can be issued multiple times to any given OTP address, but this address space can never be erased. After a given OTP segment is programmed, it can be locked to prevent further programming with the Lock Protection Registers, which are describe in Section 9.1.3.

The valid address range for OTP Program is depicted in Figure 18, "OTP Memory Map" on page 38. OTP Program operations outside the valid OTP address range will be ignored.

9.1.2 Reading OTP Data

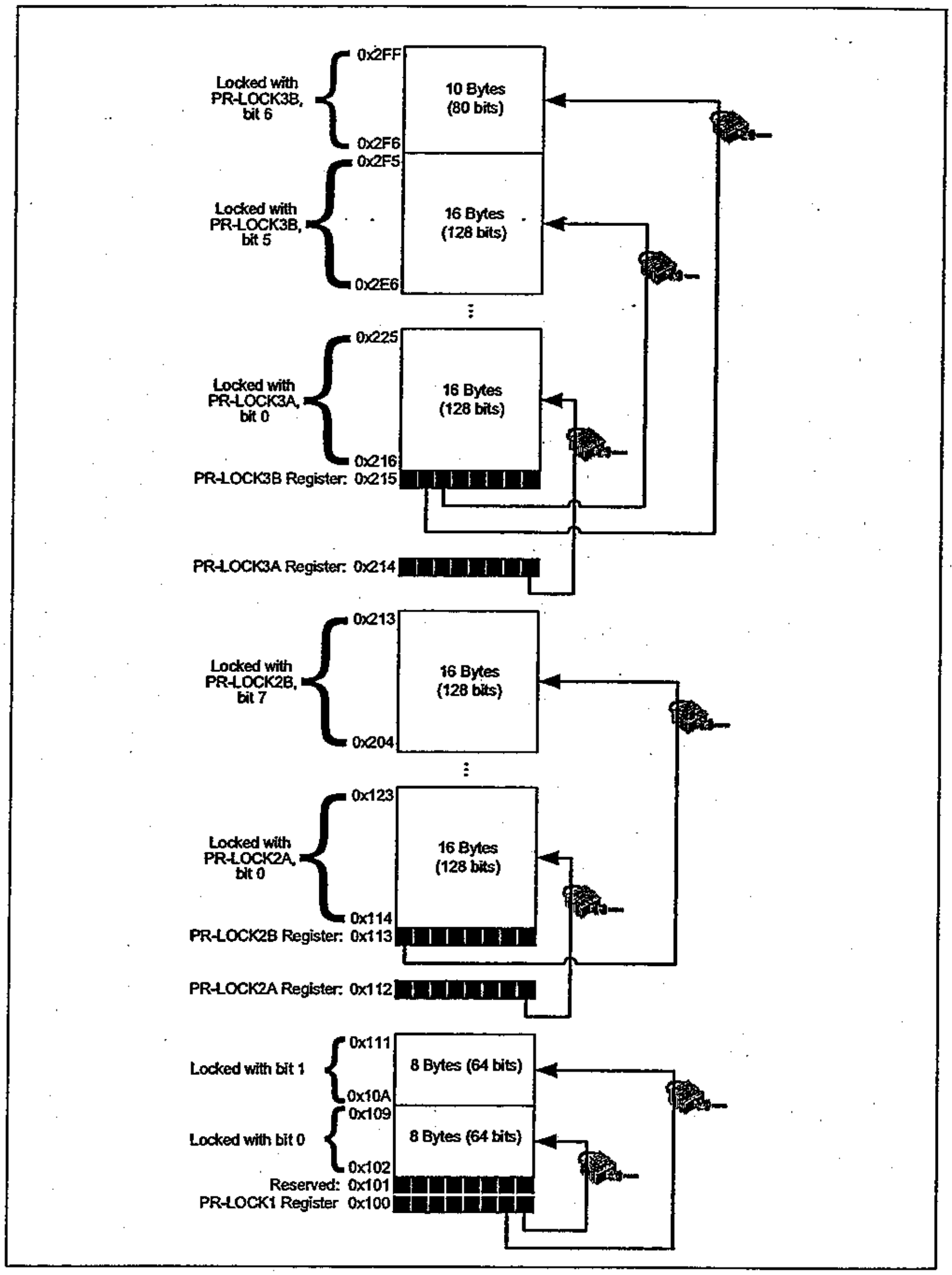
For the description and SPI protocol of the OTP Read command, refer to Table 15, "SPI Command Set" on page 26. The protocol of this command is the same as Fast Read.

The valid address range for OTP Reads is depicted in Figure 18, "OTP Memory Map" on page 38. OTP Read operations outside the valid OTP address range will yield indeterminate data.

9.1.3 Lock Protection Registers

The Lock Protection Registers (PR-LOCK1, PR-LOCK2, and PR-LOCK3) are illustrated in Figure 18, "OTP Memory Map" on page 38. PR-LOCK1 is used to permanently lock OTP addresses 0x102 through 0x111; PR-LOCK2 is used to permanently lock OTP addresses 0x114 through 0x213; PR-LOCK3 is used to permanently lock OTP addresses 0x216 through 0x2FF.

Figure 18. OTP Memory Map



Note: OTP Bytes 0x102 through 0x109 will be programmed at Intel's factory with an unchangeable, unique identifier, and bit 0 of PR-LOCK1 register will be programmed to 0.

10.0 Intel® Serial Flash Memory (S33) ID Codes

The Manufacturer ID is 89h, and the Device IDs are found in Table 19.

Through the SPI Command Set, OP code 9Fh is required. The first byte of data read is the Manufacturer ID (89h); the second byte is the upper byte of the Device ID, and the third byte is the lower byte of the Device ID.

Table 19. Device ID Codes

Device	Device ID	Mode
Intel® Serial Flash Memory (S33) 64M	8913	Bottom Boot
Intel® Serial Flash Memory (S33) 32M	8912	Bottom Boot
Intel® Serial Flash Memory (S33) 16M	8911	Bottom Boot

Appendix A Write State Machine (WSM)

Table 20 and Table 21 shows the command state transitions (Next State Table) based on incoming commands.

Table 20. Chip State and Output State Transitions (Sheet 1 of 2)

Current Chip State	Command Input to Chip and Resulting Chip Next State (1 of 2)								
	Write SR	Page Program	Read Data Bytes	Write Disable	Read SR	Write Enable	Fast Read Data Bytes	Clear SR Fail Flags	Param Block Erase
	01h	02h	03h	04h	05h	06h	0Bh	30h	40h
Ready, WEL bit =0	Ready, WEL bit = 0 (command ignored)		Ready, WEL bit does not change (Array Data Output)	Ready, WEL=0	Ready, WEL bit does not change (SR Output)	Ready, WEL=1	Ready, WEL bit does not change (Array Data Output)	Ready, WEL bit does not change	Ready, WEL bit = 0 (command ignored)
Ready, WEL bit =1	Ready. If Software Protected, SR Write occurs and WEL=0. Else, command is ignored and WEL=1. (see note 1)	If targeted address is protected, device is Ready and WEL=1. Else, device is busy and WEL=x. (see note 2)							If the targeted address is protected and a valid parameter block address, device is Ready and WEL bit = 1. Else, device is busy and WEL bit = x (see note 2)
DPD, WEL bit =x	DPD, WEL bit does not change (command ignored)								
Busy	Busy (command ignored)				Busy (SR Output)	Busy (command ignored)			

Notes:
1. Refer to Table 18 for details on HW/SW protection.
2. Refer to Table 18 for details on address protection.
3. Refer to Section 9.0, "Security Features" for details on OTP protection.
4. Refer to Table 16 for details on the Status Register.

Table 21. Chip State and Output State Transitions (Sheet 2 of 2)

Current Chip State	Command Input to Chip and resulting Chip Next State (2 of 2)								
	OTP Program	Read OTP Data Bytes	Read JEDEC ID	Release from DPD	Deep Power-Down	Bulk Erase	Sector Erase	Power Cycle (off, then on)	Program or Erase Operation Completes
	42h	4Bh	9Fh	ABh	B9h	C7h	D8h		
Ready, WEL bit =0	Ready, WEL bit = 0 (command ignored)	Ready, WEL bit does not change (OTP Data Output)	Ready, WEL bit does not change (ID Output)	Ready, WEL bit does not change	DPD, WEL bit does not change	Ready, WEL=0 (command ignored)		Ready, SR = 1Ch (see note 4)	N/A
Ready, WEL bit =1	If targeted address is protected, device is Ready and WEL bit = 1. Else, device is busy and WEL bit = x. (see note 3)					If any main memory is protected, device is Ready and WEL bit =1. Else, device is busy and WEL bit = x. (see note 2)	If targeted address is protected, device is Ready and WEL bit =1. Else, device is busy and WEL bit =x. (see note 2)		
DPD, WEL bit =x	DPD, WEL bit does not change (command ignored)				DPD, WEL bit does not change (command ignored)				
Busy	Busy (command ignored)								Ready, WEL bit = 0

Notes:
1. Refer to Table 18 for details on HW/SW protection.
2. Refer to Table 18 for details on address protection.
3. Refer to Section 9.0, "Security Features" for details on OTP protection.
4. Refer to Table 16 for details on the Status Register.

Appendix B Ordering Information

Figure 19. Ordering Information

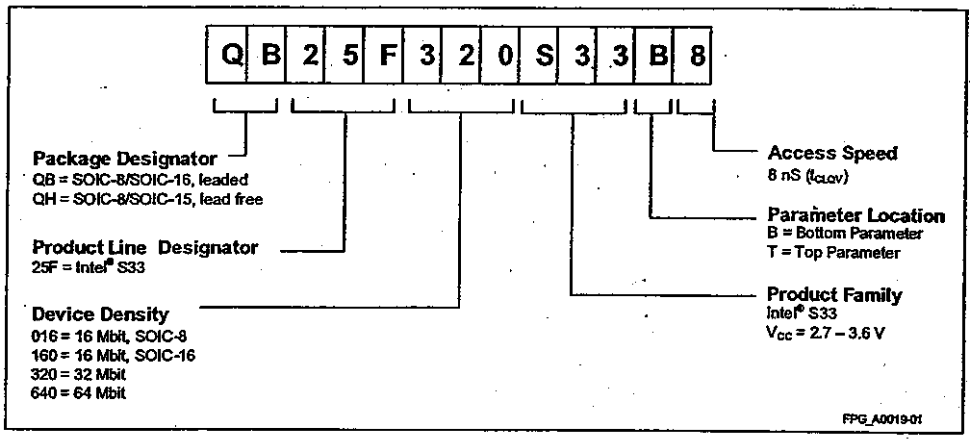


Table 22. Valid Combinations for Intel® Serial Flash Memory (S33)

32 Mbit	16 Mbit	64 Mbit
QB25F320S33B8	QB25F016S33B8/QB25F160S33B8	QB25F640S33B8
QH25F320S33B8	QH25F016S33B8/QB25F160S33B8	QH25F640S33B8

EXHIBIT D
TO WEI DECLARATION

HIGHLY CONFIDENTIAL – ATTORNEYS EYES ONLY

EXHIBIT FILED UNDER SEAL

EXHIBIT E
TO WEI DECLARATION

HIGHLY CONFIDENTIAL – ATTORNEYS EYES ONLY

EXHIBIT FILED UNDER SEAL

EXHIBIT F
TO WEI DECLARATION

HIGHLY CONFIDENTIAL – ATTORNEYS EYES ONLY

EXHIBIT FILED UNDER SEAL

EXHIBIT G

TO WEI DECLARATION

HIGHLY CONFIDENTIAL – ATTORNEYS EYES ONLY

EXHIBIT FILED UNDER SEAL

EXHIBIT H
TO WEI DECLARATION

HIGHLY CONFIDENTIAL – ATTORNEYS EYES ONLY

EXHIBIT FILED UNDER SEAL

EXHIBIT I
TO WEI DECLARATION

HIGHLY CONFIDENTIAL – ATTORNEYS EYES ONLY

EXHIBIT FILED UNDER SEAL

EXHIBIT J

TO WEI DECLARATION

HIGHLY CONFIDENTIAL – ATTORNEYS EYES ONLY

EXHIBIT FILED UNDER SEAL

EXHIBIT K

FEATURE

Published in **April/May 2006** issue of Chip Design Magazine

Flash Memory Moves from Niche to Mainstream

With the continuing growth of high-density NAND technologies, more embedded-Flash-drive devices are combining Flash media and a controller on a single chip.

By Francois Kaplan

With the massive deployment of high-speed data-transfer networks, users are being treated to an explosion of multimedia services. New devices are enabling services ranging from music phones, which are inspired by the success of MP3 players, to more general-purpose smart phones. The new multimedia-enabled handsets--and third-generation (3G) handsets in particular--have many common denominators. For instance, they all require much more memory. Users need to store both the content that they download from mobile network operators (MNOs) and their own personal data. The data may be private, such as photos and music. Or it may comprise professional e-mails and work documents. The memory in such handsets is therefore expected to reach densities of over 1 GB. Meanwhile, low-end 3G handsets will offer a bare minimum of 32 MB. Only a year ago, such requirements ranged from 16 to 256 MB.

Flash vendors have rallied to the call. They're offering a range of high-density solutions that are based on various types of technologies, manufacturing processes, and form factors. Handset designers, for their part, are faced with the non-trivial task of sorting through all of these offerings and evaluating their benefits and tradeoffs. This effort requires a basic understanding of the differences between NOR, single-level-cell (SLC) NAND, and multi-level-cell (MLC) NAND Flash--the three major technologies in use. Designers also need a more in-depth understanding of how Flash vendors are implementing these technologies. The vendors' goal is to enable handset OEMs to use them inside their newest models as quickly and cost effectively as possible.

FROM NOR TO NAND

The use of NAND-Flash technology in selected smart phones began in 2001 as an eyebrow-raising novelty. A mere handful of pioneering Flash vendors turned to NAND to meet seemingly contradictory memory requirements. They had to provide higher densities and better performance at lower costs and in smaller packages. Over the past two years, the number and proportion of NAND-based handsets has grown exponentially. Basic phones, which don't require high-density memory, continue to use NOR Flash technology. But the deployment of 3G smart phones and high-end multimedia phones, which require both high-density embedded memory and high performance, has been largely responsible for the shift from NOR to NAND.

NAND-Flash revenues exceeded those of NOR Flash for the first time in the first quarter of 2005--only three years after NAND Flash was first introduced to mobile handset manufacturers in what was until then an exclusively NOR-Flash stronghold (Semico, May 2005). This dramatic success was the result of increased memory-card sales in growing densities. It also was driven by a massive penetration of NAND as an embedded nonvolatile-memory (NVM) solution.

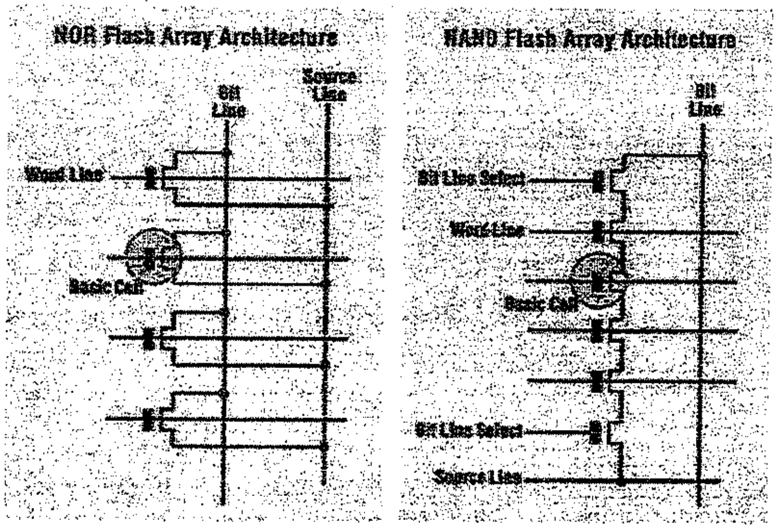


Figure 1: This figure compares the cell structures of NOR versus NAND devices.

The Table compares the key characteristics of the NOR and NAND technologies. NAND is ideal for high-density data storage, while NOR is best suited for use in code storage and execution--usually in low densities. Due to the efficient architecture of NAND Flash, its cell size is almost half the size of a NOR cell (see Figure 1). This characteristic, when combined with a simpler production process, enables NAND to offer higher densities with more memory on a given die size. The result is lower cost per MB.

NOR dominates the market in density ranges from 1 to 32 MB, while the sweet spot for NAND is 128 MB to 1 GB. (16 to 64 MB are available in small quantities and older processes.) These figures again stress the role of NOR devices for code storage and NAND devices for data storage--particularly for data-rich applications.

FROM SINGLE- TO MULTI-LEVEL-CELL NAND

NAND technology is constantly being improved to accommodate the growing demand for more embedded storage and higher performance at lower costs and in smaller packages. NAND Flash is evolving more quickly than the rate established by Moore's Law. It is doubling in density every 12 months instead of every 18 months. This evolution is being achieved both by a smaller, more precise manufacturing process (down from 130 nm in 2003 to 90-70 nm in 2005) and altered physical characteristics. The page size, for instance, has jumped from 512 B to 2 KB. The block size has gone from 32 to 128 KB. In addition to enabling higher densities, the increase in page and block sizes enhances performance. It therefore provides acceptable response rates to store more data efficiently.

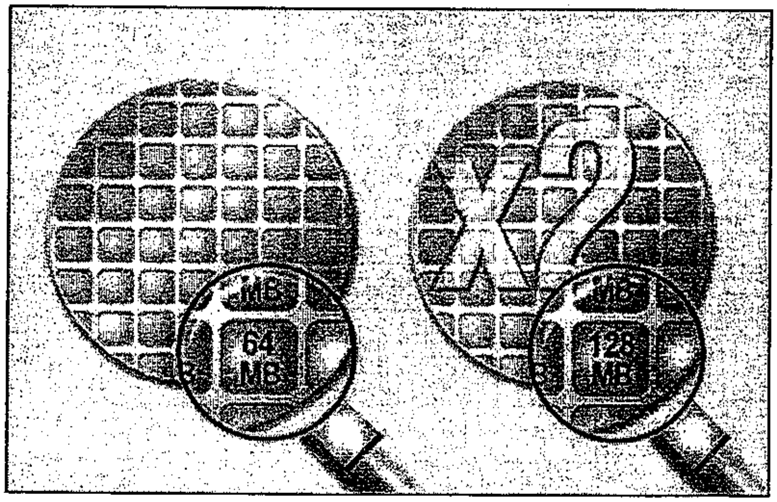


Figure 2: MLC NAND doubles Flash density in comparison to SLC NAND.

A major leap forward in NAND Flash was made with the introduction of multi-level-cell (MLC) NAND Flash in 2003. With single-level-cell (SLC) NAND, the standard 1 bit per cell was stored. In contrast, MLC NAND stores 2 bits per cell while increasing the density in a given die by up to 80% (see Figure 2). As a result, MLC NAND offers a superior cost structure. It comes with a penalty, however: Its performance and reliability are lower than SLC NAND. Despite this flaw, the performance of MLC NAND remains both significantly superior to NOR and in excess of multimedia handset requirements.

While MLC NAND represents a breakthrough in the cost structure, the technology's inherent limitations make it difficult to integrate into real-life applications:

- Slower write performance: Though read performance is similar with SLC NAND, MLC NAND delivers slower write performance.
- Lower reliability: As noted previously, NAND Flash suffers from occasional bit flips. SLC NAND can have 1 bit error per page. MLC NAND, on the other hand, can have up to 4 bit errors per page and more in the future generations. These errors occur much more frequently in NAND Flash than they do in SLC NAND. In addition, MLC NAND has a much higher level of bad blocks (up to 5%), which in turn requires a more efficient management scheme.
- Incompatible Flash management: Even the basic functionality of MLC NAND is different from that of SLC NAND. Unlike SLC NAND, for example, MLC NAND must be accessed sequentially (i.e., once a block is accessed, its pages must be filled sequentially). With SLC NAND, free pages can be written in any order. Even if write operations to a particular page with MLC NAND are successful, adjacent pages may suffer from bit flips as a result of that write operation. Higher functionality further complicates the use of MLC NAND, making it necessary to implement more advanced, Flash-management algorithms and controllers.

FROM RAW NAND TO EMBEDDED FLASH DRIVES

When NAND was first introduced to the mobile handset market, its successful implementation depended on using an Embedded Flash Drive (EFD). EFDs integrate the Flash media along with a Flash controller on the same chip. They also include specially developed Flash-management software. The very first EFD was M-Systems' DiskOnChip (also sold by Toshiba).

EFDs offer access to the NAND media through a legacy, NOR-like interface. They also provide an eXecute In Place (XIP) boot block, which makes it possible to completely remove the NOR Flash from the host system and thereby reduce the bill of materials (BOM). Additional features include more efficient power consumption, better performance, and security.

EFDs are becoming even more important as processes continually shrink, NAND-Flash qualities degrade, and the market share of complicated-to-use MLC NAND grows. EFDs deliver an easy and consistent way to implement the latest NAND technology. Memory designers aren't required to be NAND experts. Nor are they forced to compromise on using older-generation NAND components supported by chip sets.

Today, most NAND-Flash manufacturers offer EFD solutions. This trend testifies to their acceptance and success in the multimedia handset market, where they meet ever-growing data storage requirements. The EFD solutions being offered today include M-Systems' DiskOnChip, Renesas' SuperAND, Sandisk's iNAND, and Samsung's OneNAND. Newcomers to the Flash industry are allegedly planning to introduce their own EFD solutions this year.

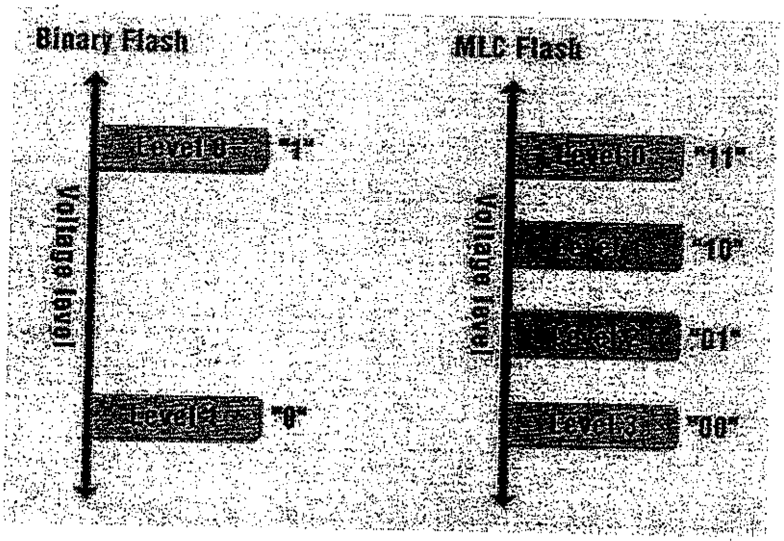


Figure 3: This comparison of different memory architectures shows the benefits of building Flash-management software into the controller as firmware.

FROM OLD TO NEW HURDLES

As NAND technology moves into mainstream and high-volume projects, the challenges for handset designers have changed. A few years ago, the issue of using NAND was mostly a technical one. No chip set supported NAND in 2001. The only way to design a NAND-based solution was by using an EFD. Today, most chip sets provide support for SLC NAND. Yet chip-set support for MLC NAND remains a challenge. In addition, two new hurdles have been spawned by the high demand for NAND technology driven by multimedia handsets and the Apple iPod nano: Flash allocation as demand exceeds supply and enabling access to the most advanced NAND technologies.

Handset vendors that use a multiple-source solution can minimize the impact of Flash allocation. In contrast, those using a single-source solution will be forced to pay higher prices. They also are likely to be allocated less supply than they need to meet customer demands. Aside from directly impacting their market share, this issue may lead to customer loss.

The issue of allocation is further complicated by the fact that advances in NAND-Flash technology and processes evolve much more rapidly than chip sets. Smart-phone chip sets, for instance, began supporting small-block SLC NAND Flash at the end of 2003. Earlier that year, both large-block SLC and MLC NAND had already been introduced. Large-block NAND was finally supported by a few chip sets by the end of 2005. Yet MLC NAND isn't expected to have chip-set support until the end of this year. Despite NAND's overwhelming acceptance in the handset market, the lower-end chip sets that target feature phones don't yet offer any NAND support.

Instead of designing a new platform for each project, most customers design a particular platform for numerous projects. The lifespan of a platform is usually about two years. Designing a platform that supports the latest NAND technology is a complex process. The most critical and time-consuming task is to update the software running on the host in order to manage the Flash. This task requires intensive testing. In many cases, designers would rather delay access to new technologies until the next-generation platform.

If the Flash-management software doesn't reside on the host, however, the integration process can be greatly simplified. New EFDs have successfully adopted this new approach. It enables them to be integrated as plug-and-play devices while using the most advanced and cost-effective NAND technologies. This new generation of EFDs offers an improved architecture: The Flash-management software is built into the controller as firmware, rather than being coupled with the host (see Figure 3). Offering an easy block-device interface minimizes the design effort for handset vendors. They can then move to the next NAND technology without changing the host design.

As the mobile market continues to offer more and increasingly sophisticated multimedia applications, the need for memory is exponentially increasing. This need is justifying the move from NOR-based to NAND-based handsets. To answer these market challenges--both technical and business--handset manufacturers would be wise to implement complete, advanced memory solutions like second-generation EFDs. By comparison, Flash components would require heavier integration efforts and higher design costs.

As NAND Flash gains industry acceptance as the best technology for today's high-density data storage, application processors are just beginning to support NAND interfaces. As the fast pace of advanced, higher-density NAND technologies continues into the foreseeable future, second-generation EFDs will have considerable added value over raw NAND media. They'll enable savvy handset manufacturers to cost effectively support new NAND technologies with plug-and-play integration ease. As a result, the manufacturers will get their new, high-density handset models to market faster.